

# Teaching Guide.

## Part-Baked Games: Chef's edition

At Digital Schoolhouse we take PEGI ratings seriously, please note that the video game used in this workshop is rated PEGI 3 - suitable for ages 3+. For more information about PEGI ratings please visit AskAboutGames <https://www.askaboutgames.com/> or the Video Standards Council Rating Board <https://videostandards.org.uk/RatingBoard/>.

## Introduction

Part-Baked Games: Chef's edition has been developed in partnership with Outright Games and introduces students to the concept of prototyping; both on paper and digitally. Learners begin by creating a paper prototype for their own version of Gigantosaurus and then a digital prototype based on these ideas. Students can use the recipe cards included in the resource pack for this workshop to aid them in building their games. They then compare this prototype to the realised title and choose one feature to add to their own game. Students then modify their game. The final stage of the workshop introduces learners to play testing and students thoroughly test their game, acting on any failed tests to fix errors.

Part-Baked Games: Chef's edition is intended as a follow on workshop from Part-Baked Games as it builds on skills students have learnt in this workshop. However, the workshop is suitable for students with any experience of using Construct 3 and the resource pack includes a basic guide that can be used to introduce new students to using the software.

# Contents

Learning Outcomes .....	3
Files/Resources.....	4
Session Overview.....	6
Digital Schoolhouse Progression Matrix.....	16
Algorithms .....	16
Programming .....	17
Data .....	18
Hardware and Software.....	19
Communication .....	20
IT.....	21
Computing Programmes of Study Links.....	22
Computational Thinking Strands .....	24

# Learning Outcomes

1. To be able to demonstrate an understanding of how audience and purpose relate to computer games
2. To be able to plan a game that considers audience and purpose
3. To be able to demonstrate the importance of correct instructions
4. To be able to write algorithms to represent player interactions within a computer game
5. To be able to create a coded solution based on a pre-written algorithm using a game engine such as Construct 3
6. To be able to create and use variables within a game
7. To be able to use selection statements within a game
8. To be able to create and use behaviours within a game
9. To be able to create and use events within a game
10. To be able to recognise what triggers an event and the action that will be taken as a result of it
11. To be able to evaluate the effectiveness of a solution compared to another one
12. To be able to identify features of a computer game
13. To be able to critically compare one solution to another and use this to identify modifications to be made
14. To be able to carry out modifications to a solution based on written feedback
15. To be able to dry run/test their game to see if it works and achieves desired results
16. To be able to de-bug a solution based on written feedback

# Files/Resources

U = Unplugged activity

Filename	Resource Type	Purpose/Description	Activity No
DSH-Teaching-Guide-Part-Baked-Games-Chefs-Edition	Teaching Guide	Detailed instructions for delivering the workshop materials including details of learning outcomes and assessment	All activities
DSH-Teaching-Presentation-Part-Baked-Games-Chefs-Edition	Teaching presentation	Teaching presentation for the delivery of this workshop	All activities
DSH-Student-Reference-Sheets-Part-Baked-Games-Chefs-Edition	Student reference sheets	Printable sheets for students to reference during their work – this includes copies of all assets and recipe cards for building various game mechanics in Construct 3  Printed one between two	All activities
DSH-Teaching-Presentation-C3-Basic-Guide	Teaching presentation	Step-by-step instructions demonstrating how create variables, events, behaviors and objects in Construct 3	7, 11
DSH-Worksheets-Part-Baked-Games-Chefs-Edition	Student worksheets	Worksheets for completion during the workshop  Printed one copy per student	All activities
PC or laptop	Activity resource	Needed for creating all Part-Baked games	7, 11
Google Chrome	Software	Needed to access Construct 3 for creating games	7, 11
Gigantosaurus_Part_Baked.c3p	Construct 3 file	Part-baked game file which includes all game assets required to create a game	7, 11

IdeasGeneratorCards	Printable resource	BAFTA YGD idea cards to help students to come up with a game idea	All activities
<a href="http://ygd.bafta.org/resources/game-idea-generator">http://ygd.bafta.org/resources/game-idea-generator</a>	Website	Online version of the BAFTA idea generator	All activities
<a href="https://editor.construct.net/">https://editor.construct.net/</a>	Website	Web-based game engine	7, 11
Coloured pens or pencils	Activity resource	Needed for completing all unplugged worksheets	U1, U2, U3, U4, U5, U6, U8, U9, U10, U12
Nintendo Switch	Games console	Needed to play the released version of the game	U8, U9
Gigantosaurus game	Computer game	Needed to compare prototype to the released version of the game	U8, U9

PLEASE NOTE: The activities outlined in this workshop pack are a suggested outline of how the workshop can be delivered. It is envisaged that teachers will adapt the resources and the organisation of them according to the needs of their class.

# Session Overview

If you are using the HTML5 version of these resources, you will find all content in a single online resource. If you intend to use the SCORM version of the resources via your VLE, it is intended that SESSION 1 should be delivered using the unplugged teaching presentation before students begin the programming tutorials in the SCORM package. Below you will find references to the separate resources available should you want to deliver the programming tutorials as stand-alone sessions.

U = Unplugged activity

## SESSION 1

Activity No	Session Content / Activity	Resources Used
	Welcome and introduction	DSH-Teaching-Presentation-Part-Baked-Games-Chefs-Edition – Slides 1 and 2
	Read ‘The Challenge’ slides to students, explaining to students that they have been commissioned to create a game based on a TV show  Definitions are provided in the notes pages of each slide for red keywords	DSH-Teaching-Presentation-Part-Baked-Games-Chefs-Edition – Slides 3 – 4
	Read the ‘Game assets’ slide to students explaining that they need to use the game assets provided as they will be creating a game for a specific audience and purpose  Definitions are provided in the notes pages of each slide for red keywords	DSH-Teaching-Presentation-Part-Baked-Games-Chefs-Edition – Slide 5
U1	Read the ‘Game idea’ slide and introduce the BAFTA YGD idea generator, depending on how you want to run this students can then either use the online version or physical cards to generate an idea for their game  NOTE: the environment is set to Jungle due to the audience and purpose of the game, you may also want to omit the	DSH-Teaching-Presentation-Part-Baked-Games-Chefs-Edition – Slides 6 - 7  IdeasGeneratorCards.pdf – printed one copy between two students and cut into cards Or <a href="http://ygd.bafta.org/resources/game-idea-generator">http://ygd.bafta.org/resources/game-idea-generator</a>

	<p>Wildcard cards as they make creating a game in some engines more difficult</p> <p>Move to the next slide 'initial ideas' and read it to your learners</p> <p>Hand out the 'initial ideas' worksheets</p> <p>Students should then complete the top bar with their game idea (either generated by the BAFTA YGD idea generator or their own idea)</p> <p>They can then move onto filling the boxes below with initial ideas for their game – encourage your students to use the coloured pens and pencils and to use the asset reference sheets to help them with ideas</p>	<p>DSH-Worksheets-Part-Baked-Games-Chefs-Edition – printed one copy per student (page 2)</p> <p>Coloured pens or pencils</p> <p>DSH-Student-Reference-Sheets-Part-Baked-Games-Chefs-Edition – printed one copy between two students (pages 2 – 4)</p>
U2	<p>Move onto the 'Game design document' slide and read it to your students</p> <p>Hand out the 'Game design document' worksheet and encourage students to fill it in – make sure that students think carefully about the audience of their game and explain what is meant by the term audience if needed</p> <p>Definition for red keywords are provided in the notes section of the presentation</p>	<p>DSH-Teaching-Presentation-Part-Baked-Games-Chefs-Edition – Slide 8</p> <p>DSH-Worksheets-Part-Baked-Games-Chefs-Edition – printed one copy per student (page 3)</p> <p>Coloured pens or pencils</p> <p>DSH-Student-Reference-Sheets-Part-Baked-Games-Chefs-Edition – printed one copy between two students (pages 2 – 4)</p>
U3	<p>Move onto the 'Screen planning' slide and read it to students, explain that students can now start to think about what their game will look like and you may like to show some examples of the different screens</p> <p>Hand out the 'Screen planning' worksheet and give students time to fill it in</p> <p>Encourage students to use the assets sheets to help them with their designs and remind them of the audience and purpose</p>	<p>DSH-Teaching-Presentation-Part-Baked-Games-Chefs-Edition – Slide 9</p> <p>DSH-Worksheets-Part-Baked-Games-Chefs-Edition – printed one copy per student (page 4)</p> <p>Coloured pens or pencils</p> <p>DSH-Student-Reference-Sheets-Part-Baked-Games-Chefs-Edition – printed one copy between two students (pages 2 – 4)</p>
U4	<p>Move onto the 'interactions' slide and explain that students are now going to plan how at least one level of their game will</p>	<p>DSH-Teaching-Presentation-Part-Baked-Games-Chefs-Edition – Slide 10</p>

	<p>work. You can remind students of the keyword algorithm here and ask them if they think planning the interactions is a form of algorithm (it is!)</p> <p>Hand out the ‘interactions’ worksheet and encourage students to use the assets sheet to design one level of their game</p> <p>Spend some time demonstrating how events work (in the form of a condition and an action) for example – if it is raining – wear a coat</p>	<p>DSH-Worksheets-Part-Baked-Games-Chefs-Edition – printed one copy per student (page 5)</p> <p>Coloured pens or pencils</p> <p>DSH-Student-Reference-Sheets-Part-Baked-Games-Chefs-Edition – printed one copy between two students (pages 2 – 4)</p>
U5 (optional)	<p>This slide is optional – if you have students that have completed their interactions worksheet they can move onto planning for progression between their levels</p>	<p>DSH-Teaching-Presentation-Part-Baked-Games-Chefs-Edition – Slide 11</p> <p>DSH-Worksheets-Part-Baked-Games-Chefs-Edition – printed one copy per student (page 6)</p> <p>Coloured pens or pencils</p>
U6 (optional)	<p>This slide is optional – if you have students that have completed their interactions worksheet they can move onto designing an icon and game box for their game</p>	<p>DSH-Teaching-Presentation-Part-Baked-Games-Chefs-Edition – Slide 12</p> <p>DSH-Worksheets-Part-Baked-Games-Chefs-Edition – printed one copy per student (page 7)</p> <p>Coloured pens or pencils</p> <p>DSH-Student-Reference-Sheets-Part-Baked-Games-Chefs-Edition – printed one copy between two students (pages 2 – 4)</p>

SESSION 2

Activity No	Session Content / Activity	Resources Used
7	<p>Move onto the 'Building your game!' section</p> <p>Hand out your students completed 'interactions' worksheets</p> <p>Read the 'Pitch' slide and explain that the paper prototype of the game has been accepted and now a digital prototype needs to be created</p> <p>Move onto the 'create your game' slide and remind students how to access and open the part-baked game file in your chosen game engine</p> <p>Demonstrate that all assets are included in the file and how to drag these into the layout area</p> <p>If your students are not used to using the game engine, you can use the Basic Guide presentation here which explains how to carry out everything included in the game mechanic recipe cards</p> <p>Move onto the 'recipe cards' slide and explain that students can use these cards to help them with implementing their game design ideas from the algorithms they planned on their interaction sheets</p> <p>Give students time to work on creating their games, we recommend that the teacher spends some time assisting students as required here</p>	<p>DSH-Teaching-Presentation-Part-Baked-Games-Chefs-Edition – Slides 13 - 16</p> <p>DSH-Student-Reference-Sheets-Part-Baked-Games-Chefs-Edition – printed one copy between two students (pages 5 - 9)</p> <p>Completed 'interactions' worksheets</p> <p>PC or laptop</p> <p>Google Chrome</p> <p><a href="https://editor.construct.net/">https://editor.construct.net/</a></p> <p>Gigantosaurus_Part_Baked.c3p file</p> <p>DSH-Teaching-Presentation-C3-Basic-Guide presentation (if required)</p>
	<p>Allow time at the end of the session to allow students to save their work</p> <p>You may need to do a demonstration to remind your students how to do this</p> <p>Slides on how to save are included in the Basic Guide</p>	<p>PC or laptop</p> <p>Google Chrome</p> <p><a href="https://editor.construct.net/">https://editor.construct.net/</a></p> <p>Gigantosaurus_Part_Baked.c3p file</p>

NOTE: most school setups mean that 'Download a Copy' will save to your students downloads folder which will require your students to move the file to a more suitable location

DSH-Teaching-Presentation-C3-Basic-Guide presentation (if required)

### SESSION 3

Activity No	Session Content / Activity	Resources Used
U8	<p>Move onto the 'Let's play some games!' section</p> <p>Move to the next slide and read the 'release' slide to your students</p> <p>Move to the next slide and read the 'Gigantosaurus' slide to your students</p> <p>Move to the next slide and read the 'Game comparison document' slide to your students, explain that your learners will now compare and contrast their prototype game with the actual released title</p> <p>Hand out the 'Game comparison document' sheets one per student and something to write with, then hand out the Switches and copies of the game</p> <p>Give your students some time to play the game and then collect the Switches back in</p> <p>Give students some time to complete their comparison sheet</p>	<p>DSH-Teaching-Presentation-Part-Baked-Games-Chefs-Edition – Slides 17 - 20</p> <p>DSH-Worksheets-Part-Baked-Games-Chefs-Edition – printed one copy per student (page 8)</p> <p>Coloured pens or pencils</p> <p>Nintendo Switch (we recommend one between two students)</p> <p>Copy of Gigantosaurus game (we recommend one between two students)</p>
U9	<p>Read students the 'released game features' slide</p> <p>Hand out the 'released game features' worksheet and then give students some time to fill in the features they liked from the released version of the game</p> <p>Encourage your students to explain why they liked each feature – you may need to encourage them to write more than 'because I did' or 'because it was fun'</p>	<p>DSH-Teaching-Presentation-Part-Baked-Games-Chefs-Edition – Slide 21</p> <p>DSH-Worksheets-Part-Baked-Games-Chefs-Edition – printed one copy per student (page 9)</p> <p>Coloured pens or pencils</p>
U10	<p>Read students the second 'interactions' slide</p> <p>Hand out 'interactions' worksheets</p>	<p>DSH-Teaching-Presentation-Part-Baked-Games-Chefs-Edition – Slide 22</p> <p>DSH-Worksheets-Part-Baked-Games-Chefs-Edition – printed one copy per student (page 10)</p>

Encourage students to choose one of the features they have identified as liking and think about how they could implement this in their own game

They should then complete a new 'interactions' worksheet, remind your learners to use the assets reference sheet

Coloured pens or pencils

DSH-Student-Reference-Sheets-Part-Baked-Games-Chefs-Edition – printed one copy between two students (pages 2 – 4)

## SESSION 4

Activity No	Session Content / Activity	Resources Used
11	<p>Move onto the 'Modify your game' section</p> <p>Hand out students completed 'interactions' sheets</p> <p>Read the 'modify' slide to your students</p> <p>Move onto the next slide and read the 'recipe cards' slide to your students</p> <p>You might need to remind your students how to open the game file that they saved earlier here – slides explaining how to open a file are included in the Basic Guide</p> <p>Give your students time to add their new feature to their game using their own 'interactions' sheets</p> <p>Use the Construct 3 basic guide if your students need a refresher on using the software</p>	<p>DSH-Teaching-Presentation-Part-Baked-Games-Chefs-Edition – Slides 23 - 25</p> <p>DSH-Student-Reference-Sheets-Part-Baked-Games-Chefs-Edition – printed one copy between two students (pages 5 - 9)</p> <p>Completed 'interactions' worksheets</p> <p>PC or laptop</p> <p>Google Chrome</p> <p><a href="https://editor.construct.net/">https://editor.construct.net/</a></p> <p>Gigantosaurus_Part_Baked.c3p file – students saved version</p> <p>DSH-Teaching-Presentation-C3-Basic-Guide presentation (if required)</p>
	<p>Allow time at the end of the session to allow students to save their work</p> <p>You may need to do a demonstration to remind your students how to do this</p> <p>Slides on how to save are included in the Basic Guide</p> <p>NOTE: most school setups mean that 'Download a Copy' will save to your students downloads folder which will require your students to move the file to a more suitable location</p>	<p>PC or laptop</p> <p>Google Chrome</p> <p><a href="https://editor.construct.net/">https://editor.construct.net/</a></p> <p>Gigantosaurus_Part_Baked.c3p file – students saved version</p> <p>DSH-Teaching-Presentation-C3-Basic-Guide presentation (if required)</p>

SESSION 5

Activity No	Session Content / Activity	Resources Used
U12	<p>Move onto the 'Testing time!' section</p> <p>Hand out the play testing sheets</p> <p>Read students the 'testing' slide and explain that testing is an important part of the process when developing games – you may like to introduce different types of testing here (unit, destructive, user acceptance)</p> <p>Students can then fill in what type of testing they will be completing in the 'purpose' box</p> <p>Explain the different columns of the play testing sheet: test – a description of the test to be carried out, expected result – what they expect will happen, actual result – what actually happened, comments – what they need to do to repair the bug if they have found an error.</p> <p>Encourage students to complete their sheets as they test their games and fix any bugs they find</p> <p>If time allows, allow students to carryout some peer testing of each other's games</p>	<p>DSH-Teaching-Presentation-Part-Baked-Games-Chefs-Edition – Slides 26 - 27</p> <p>DSH-Student-Reference-Sheets-Part-Baked-Games-Chefs-Edition – printed one copy between two students (pages 5 - 9)</p> <p>DSH-Worksheets-Part-Baked-Games-Chefs-Edition – printed one copy per student (page 11)</p> <p>Coloured pens or pencils</p> <p>PC or laptop</p> <p>Google Chrome</p> <p><a href="https://editor.construct.net/">https://editor.construct.net/</a></p> <p>Gigantosaurus_Part_Baked.c3p file – students saved version</p> <p>DSH-Teaching-Presentation-C3-Basic-Guide presentation (if required)</p>
	<p>Move to the 'game complete slide' and congratulate your students for completing their game</p> <p>Move onto 'The Challenge' slide and read out the objectives that your students have met – ask students to identify how they have completed each one</p> <p>Close the session by displaying the final title slide</p>	<p>DSH-Teaching-Presentation-Part-Baked-Games-Chefs-Edition – Slides 28 - 30</p>

The self-marking assessment for this workshop is only available in the SCORM and HTML5 versions of the tutorials

## SESSION 6

Activity No	Session Content / Activity	Resources Used
Assessment	<p>Students should complete the assessment (if completing the HTML5 version, students will be prompted to complete a teacher email and other details before starting the assessment – results will be sent to this email).</p> <p>There are 10 self-assessment questions.</p>	<p>DSH-SCORM-Part-Baked-Games-Chefs-edition</p> <p>Or</p> <p>DSH-HTML5-Part-Baked-Games-Chefs-edition</p>

# Digital Schoolhouse Progression Matrix

The Digital Schoolhouse progression matrix is a simplified mechanism for measuring pupil progress. It stretches from base level understanding at the beginner level to introducing GCSE content at the advanced level.

The shaded statements reflect skills and concepts covered in the workshop, these have been cross referenced to specific activities in order to reflect both their level of complexity and provide a guideline on which to measure progress.

For more details about this framework see 'Enter the Matrix' included in this workshop pack.

U = Unplugged activity

## Algorithms

	Beginner	Activity No	Intermediate	Activity No	Advanced	Activity No
Understanding	Understands what an algorithm is	U4, U10	Understands that algorithms are not the same as programming	U4, 7, U10, 11	Understands that different algorithms exist for the same problem	7, 11
Writing	Represents algorithms using graphical notation such as pictures	U4, U10	Represents algorithms using structured notation such as flowcharts	U4, U10	Represents algorithms using pseudocode	
Limitations	Understands that computers need precise instructions	U4, U10	Can identify tasks best completed by humans or computers		Understands that some problems cannot be solved computationally	
Planning	Can identify the steps needed to solve a problem	U4, U10	Can identify the programming constructs needed to solve a problem (pattern recognition)	U4, U10	Can identify the modules needed to solve a problem e.g. top down design	
Tracing	Uses logical reasoning to predict outputs and show an awareness of inputs	U4, U10	Uses logical reasoning to explain how an algorithm works	U4, U10	Evaluates the effectiveness of algorithms and models for similar problems	
Designing	Designs solutions (algorithms) that use sequence, selection i.e. if, then and else and iteration	U4, U10	Designs solutions by decomposing a problem and creating a sub-solution for each of these parts		Designs a solution to a problem that uses generalization to create objects and classes (OOP)	

# Programming

	Beginner	Activity No	Intermediate	Activity No	Advanced	Activity No
Writing	Can create a simple program in an environment that does not rely on text e.g. programmable robots etc	7, 11	Has practical experience of a high-level textual language, including use of standard libraries		Has experience of designing programs that include a graphical user interface	7, 11
Program flow	Understands that programs execute by following precise instructions	7, 11	Understands how modular programs work using sub-routines		Appreciates the effect of the scope of a variable e.g. a local variable can't be accessed from outside its function unless passed as a parameter	
Selection	Uses selection statements in programs, including an if, then and else statement	7, 11	Understands the difference between, and appropriately uses if and if, then and else Statements	7, 11	Uses nested selection statements	7, 11
Iteration	Uses loops, within programs	7, 11	Understands the difference between, and uses 'while', 'until' and 'for' loops	7, 11	Uses nested iteration and recursion	7, 11
Debugging	Detects and corrects simple semantic errors i.e. debugging, in programs	7, 11	Detects and corrects syntactical errors	7, 11	Applies a modular approach to error detection and correction	
Program design	Creates programs that implement algorithms to achieve given goals	7, 11	Can design a program based on an algorithm	7, 11	Designs modular programs using a range of methodologies e.g. RAD, waterfall	
Data types and structures	Declares and assigns variables	7, 11	Selects appropriate data types	7, 11	Understands and uses one and two dimensional data structures	
Operators	Uses arithmetic operators	7, 11	Uses a range of operators and expressions e.g. Boolean	7, 11	Understands and uses negation with operators e.g. not equal to	7, 11

# Data

	Beginner	Activity No	Intermediate	Activity No	Advanced	Activity No
Representation	Recognises that digital content can be represented in many forms	7, U8, 11	Understands how bit patterns represent different forms of data e.g. character sets, sound, numbers and images		Understands how the same bit patterns can be used for different forms of data e.g. metadata	
Transfer	Knows that data can be transferred from one computer to another	7, U8, 11	Knows that computers transfer data in binary		Understands and can explain the need for data compression, and performs simple compression methods	
Types	Recognises different types of data: text, number	7, U8, 11	Defines data types: string, integer, real and Boolean	7, 11	Understands how different data types can be used within data structures e.g. arrays must be made up of the same data type whereas lists can use several	
Binary	Can carry out simple binary to decimal conversions		Performs operations using bit patterns e.g. binary addition, conversion between binary and hexadecimal, binary subtraction etc		Understands the relationship between binary and electrical circuits, including Boolean logic	
File Size	Understands that data takes up space on a computer	7, U8, 11	Understands the relationship between binary and file size (uncompressed)		Knows the relationship between data representation and data quality e.g. resolution and colour depth etc, including the effect on file size	
Data and Information	Understands the difference between data and information		Recognises that poor-quality data leads to unreliable results, and inaccurate conclusions	7, 11	Understand the mechanisms used to cleanse data e.g. validation, range checks etc	
Searching	Can sort data, use filters and perform single criteria searches for information		Queries data on one table using a typical query language, including more complex searches for information e.g. using Boolean and relational operators		Queries data on multiple tables using a typical query language	
Structure	Recognises that data can be structured in tables to make it useful		Understands that all the data about a person or thing can be stored as a record		Knows what a relational database is, and understands the benefits of storing data in multiple tables	

# Hardware and Software

	Beginner	Activity No	Intermediate	Activity No	Advanced	Activity No
Processing	Understands that computers have no intelligence and that computers can do nothing unless a program is executed	7, 11	Knows that programs are executed by the processor i.e. the CPU		Understand that processors can work in different ways e.g. multitasking, scheduling	
Software	Recognises that all software executed on digital devices is programmed	7, 11	Knows that there is a range of operating systems and application software for the same hardware	7, 11	Understands the concept of proprietary and open-source software including how this relates to licencing	
Devices	Recognises that a range of digital devices can be considered a computer	U8	Understands why and when computers are used	7, U8, 11	Understands how technology has developed e.g. Moore's Law	
Components	Recognises and can use a range of input and output devices	7, U8, 11	Recognises and understands the function of the main internal parts of basic computer architecture		Knows that processors have instruction sets and that these relate to low-level instructions carried out in the main internal parts of a computer	
Operating systems	Understands that the operating system is software that specifies the function of a computing device		Understands the main functions of the operating system		Understands that there are different types of operating system and some of these common uses e.g. real time on auto pilot systems on a plane	
Data transfer	Knows that data is transferred around a computer system using input devices, sensors and application software	7, U8, 11	Knows that data can be transferred between computer systems using physical, wireless and mobile networks		Knows how data can be transferred between computer systems e.g. packet and circuit switching	
Architecture	Understands the difference between hardware and software	7, U8, 11	Understands how hardware uses software to execute instructions e.g. the fetch-execute cycle		Understands computer architecture in relation to the fetch execute cycle, including how data is stored in memory	

# Communication

	Beginner	Activity No	Intermediate	Activity No	Advanced	Activity No
WWW	Accesses content using a web browser		Understands that web pages are created using HTML and CSS		Understands how dynamic web pages use the client-server model and that web servers process and store data entered by users	
Online safety	Understands why and how to keep personal information private and knows what to do when concerned about something online		Has an awareness of a range of online harms and demonstrates responsible use of technologies and online services in order to protect themselves		Understands how and why online threats are carried out and how to protect against them	
Search engines	Navigates the web and can carry out simple web searches to collect digital content		Understands how to effectively use search engines e.g. Boolean, advanced search functions etc		Knows how search results are selected and ranked, including that search engines use 'web crawler programs'	
Networks	Understands the difference between the internet and internet service e.g. world wide web		Understands data is transmitted between digital computers over networks, including different topologies e.g. ring, star, mesh		Knows the names and purposes of network components and protocols	
Internet services	Shows an awareness of, and can use a range of internet services e.g. email		Selects, combines and uses internet services		Uses internet services to work collaboratively	

# IT

	Beginner	Activity No	Intermediate	Activity No	Advanced	Activity No
Invention	Uses software under the control of the teacher to create, store and edit digital content	7, 11	Uses and selects internet services, digital devices and application software to create, store and edit digital content	7, 11	Evaluates the appropriateness of digital devices, internet services and application software to achieve given goals	U8
Audience	Understands what an audience is	U2,	Recognises the audience when designing and creating digital content	U2, U3, U4, U5, U6, U10	Undertakes creative projects that are tailored to meet the needs of an audience	U2, U3, U4, U5, U6, U10
Purpose	Can talk about how they use computers	U2, U6	Can talk about how other people use computers	U2, U6	Can discuss the issues around how other people might use computers e.g. Data Protection Act, Computer Misuse Act, Copyright etc	U6
Evaluation	Can comment on the success of their solution	U12	Designs and uses criteria to critically evaluate the quality of solutions	U9, U12	Documents user feedback, the improvements identified, and the refinements made to the solution	U9, U12
Content	Can gather content	U1, U3, U4, U6, U9, U10	Makes judgements about content when evaluating and repurposing it for a given audience	U1, U3, U4, U6, U9, U10	Evaluates the trustworthiness of content, considers the usability of visual design features and properties of media when designing and creating digital artefacts	U1, U3, U4, U6, U9, U10

# Computing Programmes of Study Links

- 1.1 understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions
- 1.2 create and debug simple programs
- 1.3 use logical reasoning to predict the behaviour of simple programs
- 1.4 use technology purposefully to create, organise, store, manipulate and retrieve digital content
- 1.5 recognise common uses of information technology beyond school
- 1.6 use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies
  
- 2.1 design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts
- 2.2 use sequence, selection, and repetition in programs; work with variables and various forms of input and output
- 2.3 use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs
- 2.6 select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information
- 2.7 use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact
  
- 3.1 design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems
- 3.2 understand several key algorithms that reflect computational thinking [for example, ones for sorting and searching]; use logical reasoning to compare the utility of alternative algorithms for the same problem
- 3.3 use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions
- 3.4 understand simple Boolean logic [for example, AND, OR and NOT] and some of its uses in circuits and programming; understand how numbers can be represented in binary, and be able to carry out simple operations on binary numbers [for example, binary addition, and conversion between binary and decimal]
- 3.5 understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems



- 3.6 understand how instructions are stored and executed within a computer system; understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits
- 3.7 undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analysing data and meeting the needs of known users
- 3.8 create, re-use, revise and re-purpose digital artefacts for a given audience, with attention to trustworthiness, design and usability
- 3.9 understand a range of ways to use technology safely, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct and know how to report concerns
  
- 4.1 develop their capability, creativity and knowledge in computer science, digital media and information technology
- 4.2 develop and apply their analytic, problem-solving, design, and computational thinking skills
- 4.3 understand how changes in technology affect safety, including new ways to protect their online privacy and identity, and how to identify and report a range of concerns.

# Computational Thinking Strands

## AL – Algorithmic Thinking

**Ref.**      **Activity**

- |     |   |
|-----|---|
| A1  | Formulating instructions to achieve a desired effect  |
| A2  | Formulating instructions to be followed in a given order (sequence)                               |
| A3  | Formulating instructions that use arithmetic and logical operations                               |
| A4  | Writing sequences of instructions that store, move and manipulate data (variables and assignment) |
| A5  | Writing instructions that choose between different constituent instructions (selection)           |
| A6  | Writing instructions that repeat groups of constituent instructions (loops/iteration)             |
| A11 | Using an appropriate notation to write code to represent any of the above                         |
| A12 | Creating algorithms to test a hypothesis  |

## DE – Decomposition

**Ref.**      **Activity**

- |    |  |
|----|--|
| D1 | Breaking down artefacts into constituent parts to make them easier to work with  |
| D2 | Breaking down a problem into simpler versions of the same problem that can be solved in the same way (recursive and divide and conquer strategies) |

## GE – Generalisation

**Ref.**      **Activity**

G1	Identifying patterns and commonalities in artefacts
G2	Adapting solutions, or parts of solutions, so they apply to a whole class of similar problems
G3	Transferring ideas and solutions from one problem area to another

## AB – Abstraction

**Ref.**      **Activity**

Ab1	Reducing complexity by removing unnecessary detail
Ab2	Choosing a way to represent an artefact, to allow it to be manipulated in useful ways
Ab3	Hiding the full complexity of an artefact (hiding functional complexity)
Ab4	Hiding complexity in data, for example by using data structures
Ab5	Identifying relationships between abstractions
Ab6	Filtering information when developing solutions

## EV – Evaluation

**Ref.**      **Activity**

E1	Assessing that an artefact is fit for purpose
E2	Assessing whether an artefact does the right thing (functional correctness)
E3	Designing and running test plans and interpreting the results (testing)
E4	Assessing whether the performance of an artefact is good enough (utility: effectiveness and efficiency)
E5	Comparing the performance of artefacts that do the same thing
E6	Making trade-offs between conflicting demands

- E7 Assessing whether an artefact is easy for people to use (usability)
- E8 Assessing whether an artefact gives an appropriately positive experience when used (user experience)
- E9 Assessment of any of the above against the specification and set criteria
- E10 Stepping through processes or algorithms/code step-by-step to work out what they do (dry run/tracing).